



HAL
open science

Commande prédictive distribuée sous contraintes de partitionnement pour une flotte de robots mobiles

Sylvain Bertrand, Mickael Zodros

► **To cite this version:**

Sylvain Bertrand, Mickael Zodros. Commande prédictive distribuée sous contraintes de partitionnement pour une flotte de robots mobiles. SAGIP 2024, May 2024, Lyon, France. hal-04708346

HAL Id: hal-04708346

<https://hal.science/hal-04708346v1>

Submitted on 24 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Commande prédictive distribuée sous contraintes de partitionnement pour une flotte de robots mobiles *

Sylvain Bertrand¹, Mickael Zodros²

¹ DTIS, ONERA, Université Paris Saclay, Palaiseau, France
sylvain.bertrand@onera.fr

² Ecole Normale Supérieure, Rennes, France
mickael.zodros@ens-rennes.fr

Mots-clés : *Commande prédictive distribuée, Systèmes multi-robots.*

1 Introduction

Le problème considéré ici est celui de la commande distribuée d'une flotte de robots mobiles, vers une position ou une pose de référence. Ce problème peut être abordé par des méthodes de type commande en formation, où les véhicules doivent se déplacer selon un pattern de positionnements relatifs prédéfini. Ce type d'approches conduit souvent à une forte rigidité du groupe de véhicules, ce qui peut s'avérer problématique dans le cas d'une navigation en environnement encombré (présence d'obstacles). Des approches de déplacement coordonné non rigide peuvent alors être préférées.

Dans des travaux précédents [2, 4] une telle méthode a été proposée en introduisant un partitionnement de Voronoi basé sur les positions des robots. Chaque cellule de Voronoi définit une zone autour d'un robot, mise à jour en ligne, et au sein de laquelle chaque robot calcule sa propre référence locale à atteindre. La définition de cette référence locale tient compte d'une attraction vers l'objectif à atteindre pour la flotte, ainsi que d'une répulsion vis-à-vis des autres robots et des obstacles afin d'éviter des collisions. Dans [2, 4] une approche géométrique a été développée pour permettre aux robots de calculer cette référence locale de manière distribuée. Un contrôleur bas-niveau est ensuite utilisé par chaque robot pour suivre cette référence locale. L'objectif des travaux présentés ici est de développer un algorithme de commande prédictive distribuée (D-MPC) pour remplacer cette approche géométrique et le contrôleur de chaque robot. Le partitionnement de Voronoi est pris en compte par le biais de contraintes sur la position des robots au sein du problème de D-MPC. Pour ce faire, l'algorithme de *set-point tracking MPC* proposé dans [6, 5] puis étendu dans [3] est considéré comme base, et est adapté au cadre distribué considéré ici. L'algorithme D-MPC proposé permet ainsi de déterminer à la fois le point de référence local au sein de la cellule de Voronoi et la commande à appliquer par le robot.

*Ce travail a été en partie financé dans le cadre du projet ANR Delicio (ANR-19-CE23-0006).

2 Description de l'algorithme D-MPC

On considère une flotte de N_r robots mobiles, de dynamique décrite par un modèle de type unicycle représenté en temps discret par

$$x_{k+1}^i = x_k^i + T_s V_k^i \cos(\theta_k^i), \quad y_{k+1}^i = y_k^i + T_s V_k^i \sin(\theta_k^i), \quad \theta_{k+1}^i = \theta_k^i + T_s \omega_k^i \quad (1)$$

avec les notations usuelles, et où $i = 1, \dots, N_r$ désigne l'indice du robot et T_s la période d'échantillonnage. Notons $p_k^i = [x_k^i \ y_k^i]^T$ le vecteur position du robot i dans un référentiel global commun à la flotte. Supposons qu'un objectif commun à atteindre est assigné à la flotte, défini par une pose de référence $q^{ref} = [(p^{ref})^T \ \theta^{ref}]^T$ avec la position de référence $p^{ref} = [x^{ref} \ y^{ref}]^T$. A chaque instant k , chaque robot i de la flotte détermine sa cellule de Voronoi \mathbb{V}_k^i en tenant compte des positions des robots voisins (voir la description de cette procédure dans [2]). L'objectif pour chaque robot i est alors de calculer de manière distribuée un point de référence local $\bar{p}_k^i \in \mathbb{V}_k^i$ cherchant à attirer la flotte vers p^{ref} , et sa commande $u_k^i = [V_k^i \ \omega_k^i]^T$ chargée de contrôler le déplacement du robot vers \bar{p}_k^i . Ainsi, \bar{p}_k^i joue le rôle d'une référence locale que le robot i peut atteindre en toute sécurité, vis-à-vis des autres robots, car située à l'intérieur de sa cellule de Voronoi. L'algorithme D-MPC permet à chaque robot i de calculer simultanément \bar{p}_k^i et u_k^i , à partir d'informations qui lui sont localement accessibles. L'évitement de collision entre les robots et par rapport à des obstacles est également considéré par l'algorithme de commande prédictive distribuée.

Le problème d'optimisation considéré dans la D-MPC et devant être résolu à chaque instant k par chaque robot i est formulé de la manière suivante :

$$\begin{aligned} \min_{\bar{p}_k^i, \{u^i\}_k^{k+N-1}} \quad & \sum_{j=1}^N \left\| p_{k+j}^i - \bar{p}_k^i \right\|_{Q_1}^2 + \left\| \theta_{k+j}^i - \theta^{ref} \right\|_{\lambda_\theta}^2 + \sum_{j=1}^{N-1} \left\| u_{k+j}^i \right\|_R^2 + J_{reach,k}^i + J_{obs,k}^i + J_{col,k}^i \\ \text{t.q.} \quad & p_{k+N}^i = \bar{p}_k^i \\ & \bar{p}_k^i \in \mathbb{V}_k^i \\ & u_{k+j}^i = [V_{k+j}^i \ \omega_{k+j}^i]^T \in \mathbb{U}, \quad j = 1, \dots, N-1 \\ & p_{k+j+1}^i = p_{k+j}^i + T_s V_{k+j}^i \begin{bmatrix} \cos \theta_{k+j}^i & \sin \theta_{k+j}^i \end{bmatrix}^T, \quad \theta_{k+j+1}^i = \theta_{k+j}^i + T_s \omega_{k+j}^i \end{aligned} \quad (2)$$

avec

$$J_{reach,k}^i = \left(\left\| \bar{p}_k^i - p^{ref} \right\|_{Q_2} - r_{dest} \right)^2 \quad (3)$$

et les coûts associés à l'évitement de collision avec des robots voisins ou des obstacles :

$$J_{obs,k}^i = \sum_{l=1}^{N_o} \frac{\lambda^{obs}}{\epsilon + \exp \left(\left\| \bar{p}_k^i - p_l^{obs} \right\|_{P_{obs}}^2 - d_{safe}^{obs} \right)} \quad (4)$$

$$J_{col,k}^i = \sum_{l \in \mathcal{N}^i} \frac{\lambda^{col}}{\epsilon + \exp \left(\left\| \bar{p}_k^i - p_k^l \right\|_{P_{col}}^2 - d_{safe}^{col} \right)} \quad (5)$$

Le terme de coût (3) a pour objectif de diriger les robots de la flotte vers une position située à une distance r_{dest} de la référence p^{ref} . Cette formulation permet d'obtenir une "auto-organisation" des robots dans leur positionnement final autour de cette référence commune à la flotte. Les termes (4) et (5) ont pour objectif d'aider à placer le point de référence local \bar{p}_k^i au sein de la cellule de Voronoi \mathbb{V}_k^i loin des obstacles et des robots voisins, afin d'éviter les collisions. La notation \mathcal{N}^i est utilisée pour désigner l'ensemble des indices des robots voisins du robot i . Le nombre d'obstacles est noté N_o et est supposé connu des robots, ainsi que leurs positions p_l^{obs} , $l = 1, \dots, N_o$. Une contrainte terminale est considérée sur la position du robot à la fin de l'horizon de prédiction de longueur N . Des contraintes sur la commande des robots sont également prises en compte : $\mathbb{U} = \left\{ u = [V \ \omega]^T \mid V_{min} \leq V \leq V_{max}, \omega_{min} \leq \omega \leq \omega_{max} \right\}$. Les matrices $Q_1, Q_2, R, P_{obs}, P_{col}$ et les coefficients $\lambda_\theta, \lambda^{obs}, \lambda^{col}, d_{safe}^{obs}, d_{safe}^{col}$ sont des paramètres de réglage et $\epsilon > 0$.

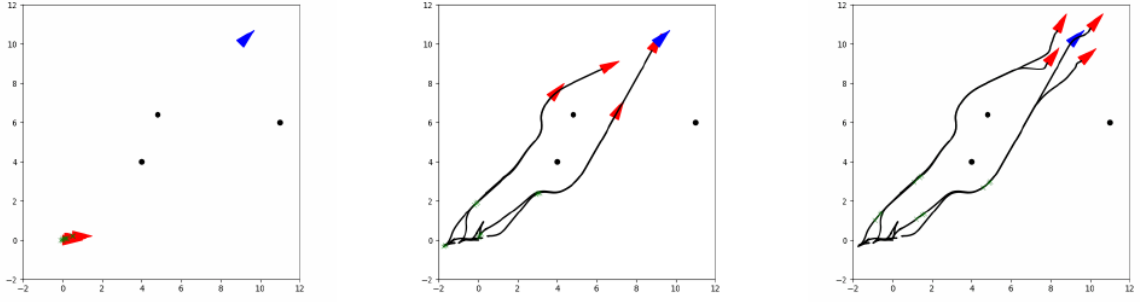


FIG. 1 – Trajectoires d’une flotte de 4 robots (flèches rouges) atteignant une pose objectif (flèche bleue) en se répartissant autour, en présence d’obstacles (points noirs).

3 Exemple de mise en oeuvre

Des résultats de simulation sont proposés en Figure 1 pour illustrer le comportement obtenu pour une flotte de $N_r = 4$ robots. Les positions initiales des robots ont été volontairement choisies trop proches les unes des autres (image de gauche). Certains robots ont alors un comportement d’attente et de déviation pour atteindre des distances de sécurité évitant les collisions. L’évitement d’obstacle est également réalisé, deux groupes de robots choisissant des chemins séparés pour les éviter par la gauche ou la droite (image au centre). Enfin, la flotte atteint la pose objectif, les robots se positionnant d’eux même autour de la position de référence à atteindre (image de droite).

4 Conclusions et perspectives

L’algorithme de commande prédictive distribuée proposé permet de déterminer la référence locale et la commande de chaque robot, en se basant sur un partitionnement de Voronoi de l’espace entre les robots. Les performances obtenues en terme de ralliement de la référence et d’évitement de collisions ont confirmé l’intérêt de l’approche, pour des temps de calcul compatibles d’une utilisation temps réel (solveur CasADi [1]). Les travaux actuels portent sur la mise en oeuvre expérimentale sur une flotte de robots mobiles.

Références

- [1] Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1) :1–36.
- [2] Bertrand, S., Sarras, I., Eudes, A., and Marzat, J. (2020). Voronoi-based geometric distributed fleet control of a multi-robot system. In *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 85–91.
- [3] dos Santos, M. A., Ferramosca, A., and Raffo, G. V. (2024). Set-point tracking MPC with avoidance features. *Automatica*, 159 :111390.
- [4] Eudes, A., Bertrand, S., Marzat, J., and Sarras, I. (2023). Distributed control for multi-robot interactive swarming using voronoi partitioning. *Drones*, 7(10).
- [5] Ferramosca, A., Limon, D., Alvarado, I., Alamo, T., and Camacho, E. (2009). MPC for tracking with optimal closed-loop performance. *Automatica*, 45(8) :1975–1978.
- [6] Limon, D., Alvarado, I., Alamo, T., and Camacho, E. (2008). MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9) :2382–2387.